Heuristics for Lifetime Maximization in Wireless Sensor Networks with Multiple Mobile Sinks

Stefano Basagni
ECE Department
Northeastern University
basagni@ece.neu.edu

Alessio Carosi and Chiara Petrioli
Dipartimento di Informatica
Università di Roma “La Sapienza”
{carosi,petrioli}@di.uniroma1.it

Cynthia A. Phillips
Sandia National Laboratories
Albuquerque, NM
caphill@sandia.gov

***Abstract*—This paper investigates heuristics to control and coordinate the concurrent movement of multiple sinks for lifetime maximization in a wireless sensor network (WSN). We have developed a centralized heuristic that runs in polynomial time given the solution to the linear program from [1] which provides a provable upper bound to the problem of controlled mobility of multiple sinks. The centralized heuristic solves the sink movement and placement problem obtaining lifetimes that are within 2% of the upper bound. We also define a deployable distributed heuristic for coordinating the motion of multiple sinks through the network. The performance comparison of our heuristics with static sink placement and with random sink mobility shows that our distributed heuristic achieves network lifetimes that are remarkably close to the optimum ones, resulting in significant lifetime improvements over random sink mobility ($+77.7\%$) and statically deployed sinks ($+382.4\%$).**

I. Introduction

Recent research stresses the importance of wireless sensor networks (WSNs) for monitoring and interacting with the physical world. Large numbers of battery-operated tiny wireless sensing devices can now be interspersed in a wide variety of environments (e.g., underground [2], underwater [3], in the wilderness [4], in the house [5], etc). They sense relevant data and transmit them to designated collection points (the *sinks*) for further relay or analysis. Given the inaccessibility of the sensor nodes and their structural and architectural simplicity, their batteries cannot easily be replaced or recharged. Therefore, energy conservation is a key issue in the design of the devices themselves, and especially in the design of the communication protocols used for data transmission from the sensors to the sinks. Given the nature of the sensor nodes, WSN protocol designers face many challenges. Among the most notorious problems, two concern static placement of the sinks. Specifically, when the sinks are statically deployed the convergence of data from multiple sensors to the sink (*data funneling*) reduces network performance considerably. Furthermore, the nodes that are closer to the sink have to act as relays for data packets from all other nodes in the network. As a consequence, their energy is soon depleted and the sink becomes unable to receive any further packet (*sink neighborhood problem*). These problems could be obviated by moving some of the network components, specifically the sinks. (A host of recent literature, surveyed in [6], shows that moving sensor nodes or relays is not as effective as moving sinks.) Many initial works on this topic [6]–[9] present analytical models as well as distributed heuristics for controlled single sink mobility that clearly show the power of moving the sink to places that are dictated by current network conditions. The results obtained by controlling the mobility of one sink are so encouraging that one wonders if and how much more improvement can be obtained by deploying multiple mobile sinks. Intuitively, having multiple data collectors will reduce the amount of data handled by a single sink, and thus the data funneling, and the rate at which the sink neighbors deplete their energy. More sinks also result in shorter routes from sensors to their closest sink. As a consequence, we expect the energy consumption and data packet latency to be lower, and the resulting network lifetime higher than when a single sink is moving.

In this paper we investigate the problem of controlling the concurrent and coordinated mobility of multiple sinks for prolonging the lifetime of a WSN. We describe a centralized heuristic that runs in polynomial time given the solution to a linear program (LP) defined in [1] which provides a provable upper bound to the problem of controlled mobility of multiple sinks. This heuristic scales to realistic network sizes and produces sink schedules that are within 2% of the upper bound provided by this LP. We also define a deployable distributed heuristic for controlling and coordinating the motion of multiple sinks through the network. In this heuristic, sinks periodically evaluate whether to move or not depending on the expected lifetime improvement that can be obtained by such a move. By gathering network status information efficiently, sinks can make this decision locally and share it with fellow sinks. So, sinks coordinate movement decisions.

Through simulations, we compare the performance of our heuristics with that of static sink placement and of random sink movements. Simulation-based experimental results show that our distributed heuristic achieves network lifetimes that are remarkably close to the optimum ones (within 25.3% of the centralized upper bound in all considered scenarios and experiments), resulting in significant lifetime improvements (from 20.6% to 77.7%) over random sink mobility. The lifetime improvements with respect to optimally placed static sinks can be as high as 382.4%. Finally, the investigation of route length and the residual energy distribution at network lifetime confirms that controlled and coordinated mobility of multiple sinks is a practical tool for significantly improving the overall performance of a WSN.

II. Problem Formulation

The scenario we consider is made up of two tiers: One that comprises a small number s of mobile sinks deployed to collect data from the sensors, and the other that contains a large number of resource-constrained sensor nodes that monitor the deployment area. The sinks, e.g., mobile robots or unmanned flying drones, roam through the network moving among a finite number of designated sink sites.[1] Mobile sinks are resource-rich nodes, i.e., they are not particularly constrained in terms of energy, mobility, computational resources, and storage. We assume that they have multiple radio interfaces so that they can communicate with the sensors and among themselves. The second tier of the network is composed of a set N of sensor nodes (or sensors for short). Each sensor $p \in N$ has an initial energy e_p. It generates data at a rate of r_p packets per second. These packets are routed either directly or via a multi-hop path to the sink sojourning at the closest site (in terms of number of hops) according to a given routing protocol. Sensor p requires α_p joules per packet for sensing, creating, and transmitting packets it generates *locally*. It requires β_p joules per packet to *receive* and *relay* a packet for another sensor.

Since the sinks are mobile, the sensors must learn which is the closest sink that is available to receive their packets. More specifically, at any time, a sink is either at a sink site or it is moving. If it is at a sink site, it can be either *active*, meaning it is available to receive sensor data, or *inactive*. When a sink at a sink site becomes active, it must broadcast its availability to the sensors. Similarly, when it becomes inactive, it must broadcast its unavailability. These broadcasts require energy from the sensors and are sink-site dependent. An active sink must deactivate before moving. We require that at least one sink is active at all times. This design choice allows the sensors to always have a destination for their packets. This has the benefit of avoiding high packet end-to-end latencies (nodes do not need to buffer packets while the sinks are moving).

Our objective is to find a schedule of sink movements that maximizes the network lifetime, which is reached when the first sensor runs out of battery power.

A more formal description of our problem follows. Let V be the set of possible sink sites and let C be a set of all possible sink configurations. A configuration is a subset of $1 \leq k \leq s$ sites in V where there are active sinks available to receive sensor data. Let y_{pwc} be the percentage of traffic sensor $p \in N$ sends to a sink at site $w \in V$ when the sink configuration is $c \in C$. When sensor $p \in N$ is sending some traffic to sink site $w \in V$, let ρ_{pqw} be the percentage of that p-to-w traffic sent through sensor $q \in N$. This allows arbitrarily complex routing strategies provided each sensor decides where to send data and how to route it based only on the sink configuration. In configuration c, the ρ_{pqw} and y_{pwc} determine the routes for the data that sensor p generates, and hence the energy intermediate nodes q consume relaying p's data.

The sink configuration changes whenever a sink becomes active or inactive, triggering a reconfiguration broadcast for each sink that changes status. Suppose we wish to transition from configuration $c_1 \in C$ to $c_2 \in C$. Because we require at least one active sink at all times, we may require intermediate configurations to make the transition. We call configurations we wish to use for a while, such as c_1 and c_2, *major* configurations, and we call intermediate configurations used only to move between major configurations *transient* configurations.

We require that transient configurations hold for t_{trans} seconds and that major configurations hold for $t_{min} \geq t_{\text{trans}}$ seconds. For example, we might prefer to hold major configurations longer to allow routes to stabilize and run for enough time to justify the configuration cost, or we may wish to minimize sink movement for other reasons (e.g., a fragile environment). By varying the size of t_{min}, one can explore the trade-offs between sink mobility and network lifetime.

Our goal is to determine an ordered set of major configurations $C_m = c_0, c_1, \ldots, c_\ell$ and a time $t_i \geq t_{min}$ for each selected configuration c_i. We are allowed to insert transient configurations with $t_i = t_{\text{trans}}$ in order to provide a most direct transition between two major configurations. Let C_t be the set of transient configurations. Every sensor must have enough initial energy to support all the selected major and transient configurations for the selected amount of time and to support all the reconfiguration (activation/deactivation) broadcasts from the sinks. We wish to find a schedule that maximizes the network lifetime: $\sum_{i \in C_m \cup C_t} t_i$.

III. Centralized heuristic

Basagni et. al. [1] define a linear program that provides an upper bound on the maximum lifetime. The solution to the LP provides a time $t^*_c \geq 0$ for a polynomial number of configurations $c \in C$, where $t^*_{c\prime} = 0$ for all the configurations $c\prime \notin C$. Ignoring reconfiguration broadcast costs and assuming configurations can change instantaneously, then each configuration c can run t^*_c time before the first sensor fails. This provides an (unachievable) upper bound on the lifetime of a centralized heuristic. Our centralized heuristic solves the full problem, with correct configuration transitions and broadcast costs. Given the solution to the LP, the high-level operations for the centralized heuristic are as follows:

1. Let $B = t_{min}$.
2. Let $\mathcal{C} \subseteq C = \{c \in C | t^*_c \geq B\}$. That is, we select the set of configurations for which the LP assigns a time of at least B (initially t_{min}).
3. Order the configurations in $\mathcal{C}$, preferably with more closely-related configurations near each other.
4. Compute transient configurations between each pair of adjacent configurations.
5. Solve another final LP (LPF) to adjust the times for each configuration, enforce minimum times on configurations, and account for sink-movement broadcast costs.

[1] The case where the sink can sojourn anywhere in the deployment area easily reduces to the case of a finite set of sink sites given that the area can be partitioned into regions where sinks' neighbors do not change.

6. If LPF is infeasible, increase B (e.g., to allow removal of the next-shortest configuration) and return to step 2.

Before describing the steps in more detail, we first consider transient states. Because sinks require time to move between sink sites and because we require at least one active sink at all times, we forbid certain types of transitions that appear to "teleport" sinks. A transition between configuration $c_1 \in C$ to configuration $c_2 \in C$ is *legal* if $|c_2 - c_1| \leq s - |c_1|$. That is, the number of sites that receive new active sinks in the new configuration must not exceed the number of inactive sinks in the preceding configuration. The inactive sinks can move to the new sites during the time configuration c_1 holds.

We can transition between any pair of configurations using at most two intermediate configurations. Suppose we wish to transition from configuration $c_1 \in C$ to $c_2 \in C$. If either is a subset of the other, we can transition directly by (de)activating the appropriate set of sinks. If c_1 and c_2 have a site in common or either has fewer than s sites, then we need only one transient configuration. It contains the common (unchanging) sites plus some new (c_2) sites to be occupied by sinks inactive in c_1 plus some old (c_1) sites occupied by sinks that will be be inactive in c_2. We need two configurations only when c_1 and c_2 both have s sites with none shared. In the first transient configuration, a subset of sinks from c_1 move while the others remain active. In the second transient configuration, the traveling sinks arrive at a subset of the new sites while those from the first transient state deactivate to move.

We now consider each step of the centralized algorithm, starting with step 3. We use a simple traveling salesperson (TSP) model. Two (ordered) configurations $c_i, c_j \in C$ have distance 1 plus the minimum number of intermediate configurations needed to implement the transition. We wish to find a traveling salesman path (not a closed tour) among the chosen configurations. The optimal TSP minimizes the number of intermediate configurations we must add, which heuristically minimizes the amount of time spent in these added configurations. These are tiny and easy problems for the free TSP code Concorde [10]. However, one could also use a polynomial-time approximation algorithm for TSP such as Christofides' heuristic [11].

In step 4, suppose we wish to compute the transition states between two states c_i and c_{i+1} that are adjacent in the ordering computed in step 3. We concentrate on the cases that require 2 transient steps, since the other cases are almost always completely determined. There are many possible pairs of intermediate states. We must choose which sinks move first, and where they move to. The initial LP chooses a set of configurations for which the vectors of energy costs for each sensor (weighted by t_i^*) pack well into the vector of initial sensor energies. Ideally, we would like the transient configurations to pack well as a group. However, this would be a difficult vector optimization problem. Instead, we try to keep the transient configurations between a pair of major configurations as close to these major configurations as possible. We hope, then, that they will pack well, as the major configurations do.

To create the transient configurations between major configurations c_i and c_{i+1}, we first consider configuration c_i. We estimate the routing distance between each pair of sink sites in c_i. We then find a minimum-weight maximum-cardinality matching in the complete graph with a node for each site in c_i, and edges weighted by this pairwise distance. We then pick an element from each matched pair arbitrarily and move this set of sinks M. If there are an odd number of sinks, the unmatched sink can move either first or second. Our hope is that if sites v_i and v_j are matched, the nodes sending to v_i will be instead redirected to v_j (and vice-versa) maintaining a nodal energy consumption approximating the consumption in configuration c_i. We then compute a similar matching in c_{i+1} and use that to pick the new locations L for the sinks from sites M to move to. So the transient configurations are $c_i - M$ and L.

We now consider the final LP. Because we have selected the precise set of configurations (steps 2 to 4), we now no longer have to allow for zero values of t_c. So we can enforce minimum times for configurations. Because we know the order of the configurations, we can account for broadcast costs. Specifically, each sink in the initial configuration c_1 must broadcast its activation. Moving between c_i and c_{i+1}, all sink sites in $c_i - c_{i+1}$ must broadcast their deactivation and all sinks in $c_{i+1} - c_i$ must broadcast their activation. Each broadcast from a sink site $w \in V$ has a potentially different energy cost for each sensor. Let γ_p be the total energy cost for sensor p for all broadcasts associated with the specific sequence of configurations. Let $\mathcal{C}_t$ be the set of transient configurations we compute in step 4. Then the final LP, called LPF, is as follows:

$$\text{maximize} \sum_{c \in \mathcal{C} \cup \mathcal{C}_t} t_c$$

subject to:

$$\sum_{c \in \mathcal{C} \cup \mathcal{C}_t} \left(\alpha_p r_p t_c + \sum_{q \in N, w \in V} r_q \beta_p \rho_{qpw} y_{qwc} t_c \right) \leq e_p - \gamma_p$$

$$\begin{aligned} & \forall p \in N \\ t_c \geq t_{min} \quad & \forall c \in \mathcal{C} \\ t_c = t_{\text{trans}} \quad & \forall c \in \mathcal{C}_t. \end{aligned}$$

The first constraints require all sensors to be alive throughout the network lifetime. Given a sensor $p \in N$ and a configuration $c \in C \cup C_t$, the expression in the parentheses on the left side of the constraint for sensor p gives its energy consumption while the system is in configuration c. The right-hand side is the total energy sensor p has after all broadcast energy costs. The second and third constraints guarantee that the major configurations hold for at least t_{min} seconds, and intermediate configurations for exactly t_{trans} seconds.

We expect our centralized heuristic to work well when t_{min} is significantly greater than t_{trans}. In this case, the transitions between major configurations are relatively minor compared to the time major configurations hold. Intuitively, given an ordering of major configurations, we slice pieces out of the start/end of the major configurations to insert the transitions.

IV. Distributed Heuristics

We consider mobile sinks as resource-rich nodes, in that they have no constraints in terms of energy, mobility, computational resources, storage and communications. The sinks are multi-radio nodes capable of communicating with the sensors for data collection and among themselves for sharing their view of the network. The $s > 1$ sinks perform an initial training at network set up. They travel to each sink site and flood to all sensors the request to return test packets according to the specific routing protocol in use. Each test packet carries information about the route followed from its source to the sink at that site. (This training can be performed by one sink, or sinks can divide the sites among themselves, and eventually they share the collected information.) After receiving a few test packets, a sink sojourning at site w is able to estimate, for each node p, the fraction of packets generated by source node q that will be relayed by p when q transmits to a sink at that site. (i.e., the sink is able to estimate the ρ_{qpw}). During the training phase, sensor nodes also learn other useful information (e.g., the hop distance from the sink). They send this information to the sink in the test packet, so the trained sink can determine how each sensor will partition its traffic among the active sinks in each possible configuration.

After the training phase, each of the s sinks chooses a unique site in V and floods packets to the nodes advertising its current location, thus allowing nodes to set up routes. Node energy is divided into levels. When a node decreases its energy from a level to a lower one, it piggybacks this information on a data packet that is then sent to the closest sink. This allows the sink to estimate the residual energy of the sensors sending packets to it. A sink can also estimate the data rate of each sensor transmitting to it. The sinks share this information with each other. Periodically (i.e., every t_{min}) each sink decides whether to move or not. Before doing so, it waits for a random time $\ll t_{min}$ to de-synchronize sink decisions. After this time, based on its current information about the status of the network (energy level of the sensors, the traffic they handle and which sink is currently at, or traveling to, what site) the sink computes the expected network lifetime obtainable by moving to an unoccupied site. If sites exist such that moving to one of those sites would extend the network lifetime more than δt_{min} beyond the lifetime obtainable by staying at the current site, the sink performs the following operations: 1) It communicates to the sensors currently reporting to it that it is on the move, shutting down the routes to its current site (*sink deactivation*); 2) It chooses the site moving to which induces the maximum expected network lifetime; 3) It communicates to all the other sinks that it has decided to move to the selected new site; 4) Moves to the new site, and 5) Upon arriving at the new site sends a routing packet for establishing new routes from nearby sensor nodes to the site (*sink activation*).

For the purpose of comparing our distributed heuristic with a solution corresponding to random mobility we also consider the following sink mobility scheme, termed RND: Every t_{min} seconds each sink decides whether to move or not, selecting the next site among all possible unoccupied sites and its current site randomly. A sink communicates its decision to its peers, so that no two sinks go to the same site. Route management happens as for our distributed heuristic.

V. Performance Evaluation

We compare the proposed centralized (CEN) and distributed (DIS) heuristics to an upper bound for an optimal solution (OPT) [1], to random mobility (RND) and to the case where sinks are static and optimally placed (STATIC). We consider the following realistic scenarios: 400 wireless sensor nodes are deployed on a 20×20 grid over a square deployment area of side $L = 475$m. Each node transmission radius is 25m, so each node has at most 4 neighbors. The initial energy of each sensor is 50J. Nodes generate packets of 512B at a rate of 0.5bps. These packets are sent to the closest sink using (hop-based) shortest path routing. The channel data rate is 250Kbps. The transmission power and the receiving power are 0.0144W and 0.0125W, respectively, according to the specifications of the TR 1000 radio transceiver from RF Monolithics. Sinks are free to move from any of the sites of a 4×4 and 8×8 grid to any other site of the grid. We vary the number of sinks in the range $[2, 8]$. Protocol related parameters are configured as follows. The parameter t_{min} ranges in the set $\{50, 100, 250\}$Ks. Parameter t_{trans} is equal to 10Ks. The threshold δ that governs a sink movement decision is set to 0.1. Nodal energy is partitioned into 40 levels, each being 2.5% of the initial energy. Our results consider the energy nodes expend during the sink training phase as well as that needed for route maintenance, and sink activation/deactivation. We ran 100 experiments for each displayed value, which achieves a 95% confidence level within a 5% precision.

Our experiments concern the following metrics: 1) Network lifetime, defined as the time till the "death" of the first sensor because of energy depletion; 2) Distribution of the nodal residual energy at lifetime, and 3) route length. We consider how each is affected by different mobility schemes and by the number of mobile sinks.

Tables I and II show the network lifetime (in millions of seconds) induced by the various protocols when varying t_{min}, the number s of sinks, and the number of sink sites. Each table entry shows the absolute lifetime and the percentage increase with respect to OPT (in parenthesis). We observe that the centralized heuristic CEN achieves network lifetimes that are remarkably close to the optimum. The gap from OPT is always below 2%. Our centralized heuristic starts from the set of (good) configurations that are produced by OPT. We observe that OPT spends the majority of time on some of these configurations which are particularly effective in balancing the traffic among the sinks, reducing the route length, and thus the overall network load and the rate at which nodes (especially those around the sinks) consume energy. These configuration combinations stress the network nodes quite evenly. CEN uses these same configurations adding intermediate ones. It selects the times the sinks spend in each configuration by solving LPF which forces each selected configuration to last

Table I
LIFETIME (% GAP FROM OPT), 4×4 GRID

s	t_{min}	OPT	CEN	DIS	RND	STATIC
2	50K	46.71	46.6 (0.2)	44.1 (5.5)	29 (37.9)	11.1 (76.2)
	100K	46.71	46.6 (0.2)	43.8 (6.2)	28.8 (38.3)	11.1 (76.2)
	250K	46.71	46.5 (0.4)	43.4 (7)	27.4 (41.3)	11.1 (76.2)
3	50K	61.14	61 (0.2)	54 (11.6)	38.2 (37.5)	14.8 (75.8)
	100K	61.14	61 (0.2)	53.3 (12.8)	37.6 (38.5)	14.8 (75.8)
	250K	61.14	60.9 (0.4)	52.1 (14.7)	35.4 (42.1)	14.8 (75.8)
4	50K	75.94	75.8 (0.1)	58.5 (22.9)	45.6 (39.9)	19.1 (74.8)
	100K	75.94	75.8 (0.1)	57.9 (23.7)	44.7 (41.1)	19.1 (74.8)
	250K	75.94	75.7 (0.3)	57.8 (23.8)	42.2 (44.4)	19.1 (74.8)
5	50K	82.42	82 (0.5)	62.9 (23.6)	50.8 (38.3)	22.3 (72.9)
	100K	82.42	82 (0.5)	62.4 (24.2)	50.2 (39)	22.3 (72.9)
	250K	82.42	81.9 (0.6)	61.5 (25.3)	48.5 (41.1)	22.3 (72.9)
6	50K	84.97	84.9 (0.1)	67.9 (20)	55.6 (34.5)	28.8 (66.1)
	100K	84.97	84.9 (0.1)	67.5 (20.5)	55 (35.2)	28.8 (66.1)
	250K	84.97	84.9 (0.1)	67.3 (20.7)	53.7 (36.8)	28.8 (66.1)
7	50K	87.29	87.2 (0.1)	73.2 (16.1)	60.2 (31)	33.7 (61.3)
	100K	87.29	87.2 (0.1)	72.9 (16.4)	59.7 (31.6)	33.7 (61.3)
	250K	87.29	87.2 (0.1)	72.4 (17)	58.1 (33.4)	33.7 (61.3)
8	50K	88.96	88.9 (≈0)	76.5 (14)	63.4 (28.7)	45.2 (49.2)
	100K	88.96	88.9 (≈0)	76.1 (14.4)	63.1 (29)	45.2 (49.2)
	250K	88.96	88.9 (≈0)	75.4 (15.2)	61.6 (30.7)	45.2 (49.2)

Table II
LIFETIME (% GAP FROM OPT), 8×8 GRID

s	t_{min}	OPT	CEN	DIS	RND	STATIC
2	50K	79.51	78.8 (0.9)	68.5 (13.8)	39.2 (50.7)	14.2 (82.1)
	100K	79.51	78.8 (0.9)	67.8 (14.7)	39.2 (50.7)	14.2 (82.1)
	250K	79.51	78.4 (1.4)	64 (19.5)	38 (52.2)	14.2 (82.1)
3	50K	105.9	105.2 (0.6)	90.1 (14.9)	50.7 (52.1)	20.8 (80.3)
	100K	105.9	105.3 (0.5)	89.3 (15.6)	50.7 (52.1)	20.8 (80.3)
	250K	105.9	105.2 (0.6)	87.6 (17.2)	49.3 (53.4)	20.8 (80.3)
4	50K	131.4	130.6 (0.6)	106.4 (19)	63.4 (51.7)	27.7 (78.9)
	100K	131.4	130.6 (0.6)	105.7 (19.5)	63.6 (51.6)	27.7 (78.9)
	250K	131.4	130.5 (0.6)	102.5 (22)	61.5 (53.2)	27.7 (78.9)
5	50K	150.1	149.2 (0.6)	120 (20)	75.6 (49.6)	34.3 (77.1)
	100K	150.1	149.2 (0.6)	118.8 (20.8)	75.9 (49.4)	34.3 (77.1)
	250K	150.1	149.2 (0.6)	117 (22)	73.9 (50.7)	34.3 (77.1)

at least t_{min}. The time spent by the sinks in each major configuration might, in principle, differ substantially from corresponding times in OPT. However, we notice that those good configurations where OPT sends the sinks for the most time are also those where CEN spends the vast majority of time. This is the natural consequence of OPT and CEN being optimization formulations. The LPF in CEN optimizes the lifetime and deems useful to spend long times in those good configurations where traffic is delivered with low energy consumption and the energy toll is balanced among the nodes, i.e., the good configurations identified by OPT. The additional configurations in which CEN spends also a significant amount of time are then properly selected not to stress on areas of the networks where the energy has already been depleted by the main configurations, and to instead drain energy from all the regions at high residual energy in the network. This justifies the near-optimal performance of the centralized heuristic.

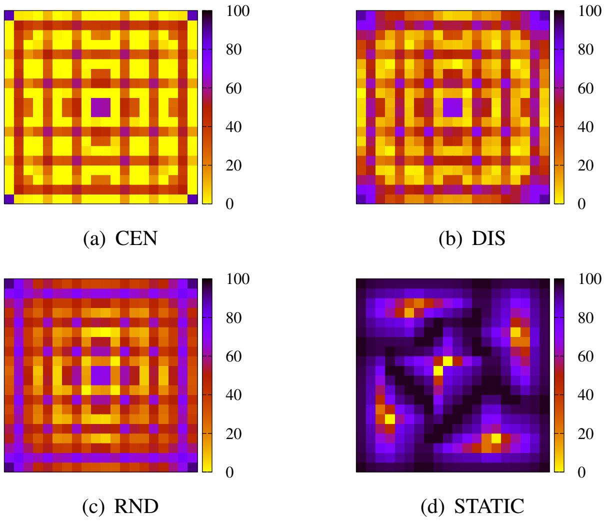
The distributed heuristic DIS is aware of the residual energy at all the network nodes because of the exchange of information among the sinks. Based on this (approximate) knowledge and the nodal data rate knowledge, DIS selects configurations that balance network load and nodal energy consumption. This traffic and energy-aware approach pays off in terms of network lifetime, which is always within a 25.3% gap from OPT's lifetime. Moreover, the improvement with respect to STATIC is as high as fourfold, which stresses the goodness of exploiting sink mobility.

Sink mobility is advantageous even with strategies like RND that use no nodal status information. Random movements of the sinks double the network lifetime with respect to STATIC.

We also observe that OPT and the other heuristics induce longer lifetimes when the number of sinks increases. This is because the network traffic is partitioned among a larger set of sinks, sink neighbors receive fewer packets, routes are shorter and the overall energy consumption is lower. However, we notice lifetime improvements which are not linear in the number of sinks. In other words, having two sinks does not double lifetime compared to one sink. Having three sinks does not triple lifetime, and so on. This is evident in Table I, which shows that deploying 8 sinks only leads to a 17% improvement in OPT lifetime with respect to scenarios with 4 sinks. Given our set of sink sites, we do not expect that linear improvement is possible. The route length and the overall energy consumption do not halve when we double the number of sinks. In addition, achieving linear improvements would require sinks to be assigned (on average) to configurations which perfectly partition data sources to the sinks. Such perfect configurations are rare, if possible at all. Moreover, if it were possible for sinks to transition only among this kind of node-balanced configurations it would be challenging to obtain good energy balancing. Energy balancing is in fact the consequence of the fine tuning of the time spent by the sinks in different configurations (including unbalanced ones) so that all the nodes relay a similar amount of traffic over time and the energy consumption is minimized.

For all protocols, increasing the number of sink sites remarkably increases lifetime. For example, when 5 sinks can visit 64 sink sites the lifetime is 150.1Ms. When restricted to 16 sink sites instead the lifetime is 82.42Ms. More sites allows the protocol to choose among more configurations. Denser sink sites allow the sinks to drain energy from all the different areas in the network. We finally observe that both DIS and RND produce higher lifetime increases as t_{min} decreases. (This is particularly evident for RND.) The reason is that higher t_{min} results in a coarser selection of the times spent in the various configurations, and therefore in a worse energy balancing. In addition, the toll to pay for having entered an "unfortunate" configuration is paid for a longer time.

Beyond achieving improved network lifetime CEN and DIS yield a more even distribution of nodal residual energy compared to RND and STATIC (confirming they are able to better distribute the load over different nodes over time). Ideally, we would like the sinks to coordinately move, changing their positions over time so that the energy is drained from all the areas in the networks. CEN and DIS satisfactorily achieve our goal. This is clearly shown in Figure 1 which displays the residual energy of the nodes at network lifetime for a



(a) CEN

(b) DIS

(c) RND

(d) STATIC

Figure 1. Residual energy at lifetime with 5 sinks in a 8 × 8 grid

given run in scenarios when 8 sinks can select among 64 sites (performance does not significantly change when considering different runs). A lighter color means a lower percentage of residual energy in that area of the network. We observe that CEN results in better load balancing that DIS, which in turn improves over RND and STATIC. More precisely, at lifetime CEN shows an impressive percentage of nodes with very little energy left, a witness of its good energy drainage balancing property. The fraction of nodes with less than 20% (40%) residual energy is 52.5% (80%) in CEN. In DIS the percentage of nodes with less than 20% of the initial energy at lifetime is around 27%. The percentage of nodes with less than 40% of the initial energy is 53%. These figures reduce to 5.41% and 23.7% for RND, and to 1.75% and 3.5% for STATIC. Mobility scheme and number of sinks both affect route lengths. Tables III and IV display the average length of the routes traversed by data packets. Increasing number of sinks corresponds to shorter route lengths. This suggests that in addition to increasing the network lifetime, deploying multiple sinks enables lower latencies and decreased data funneling.

Table III
ROUTE LENGTH FOR SINKS IN A 4 × 4 GRID

s	CEN	DIS	RND	STATIC
2	10.05	13.14	15.24	8.85
3	8.16	10.49	12.14	6.89
4	6.90	9.22	10.43	6.58
5	6.25	8.08	9.15	5.63
6	5.71	7.63	8.32	5.61
7	5.18	6.92	7.64	4.89
8	4.75	6.47	7.11	4.41

VI. Conclusions

This paper explores ways of deploying multiple mobile data collectors for lifetime improvements in wireless sensor networks. We defined two heuristics that solve the problem of maximizing network lifetimes producing results remarkably close to the optimal ones. Through experiments conclude that coordinated and controlled mobility of the sinks is always advantageous, yielding remarkable lifetime improvements over all considered cases.

Table IV
ROUTE LENGTH FOR SINKS IN A 8 × 8 GRID

s	CEN	DIS	RND	STATIC
2	9.51	11.4	13.43	8.21
3	7.39	8.78	10.84	6.27
4	6.19	7.23	9.37	5.15
5	5.43	6.57	8.19	4.66

VII. Acknowledgments

This work was partially supported by NSF grant #0738720, by the FP7 EU project "SENSEI, Integrating the Physical with the Digital World of the Network of the Future," Grant Agreement Number 215923, www.ict-sensei.org, and by the International FIRB contract RBIN047MH9. Sandia is a multipurpose laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. The authors are grateful to Bob Carr of Sandia National Laboratories for useful comments.

References

[1] S. Basagni, C. Carosi, C. Petrioli, and C. A. Phillips, "Moving multiple sinks through wireless sensor networks for lifetime maximization," *Poster at IEEE MASS 2008*, Atlanta, GA, September 29–October 2 2008.

[2] I. F. Akyildiz and E. P. Stuntebeck, "Wireless underground sensor networks: Research challenges," *Elsevier's Ad Hoc Networks*, vol. 4, no. 6, pp. 669–686, November 2006.

[3] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges," *Elsevier's Journal of Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, May 2005.

[4] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *Proceedings of the* 10*th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS-X*, San Jose, CA, October 5–9 2002, pp. 96–107.

[5] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. A. Stankovic, "An assisted living oriented information system based on a residential wireless sensor network," in *Proceedings of the* 1*st Distributed Diagnosis and Home Healthcare (D2H2) Conference*, Arlington, VA, April 2–4 2006, pp. 95–100.

[6] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, "Controlled sink mobility for prolonging wireless sensor networks lifetime," *ACM/Springer Wireless Networks*, vol. 14, no. 6, pp. 831–858, December 2008.

[7] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in *Proceedings of IEEE Globecom 2003*, vol. 1, San Francisco, CA, December 1–5 2003, pp. 377–381.

[8] J. Luo and J.-P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings of IEEE Infocom 2005*, vol. 3, Miami, FL, March 13–17 2005, pp. 1735–1746.

[9] A. P. Azad and A. Chockalingam, "Mobile base station placement and energy aware routing in wireless sensor networks," in *Proceeding of the IEEE Wireless Communications and Networking Conference, WCNC 2006*, vol. 1, Las Vegas, NV, April 3–6 2006, pp. 264–269.

[10] "Concorde TSP solver," 2005, http://www.tsp.gatech.edu/concorde.html.

[11] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," Carnegie-Mellon University, Graduate School of Industrial Administration, Tech. Rep. 388, 1976.